UNCLASSIFIED

Defense Technical Information Center Compilation Part Notice

ADP010333

TITLE: Unmanned Research Vehicle [URV]: Development, Implementation, & Flight Test of a MIMO Digital Flight Control System Designed Using Quantitative Feedback Theory

DISTRIBUTION: Approved for public release, distribution unlimited

This paper is part of the following report:

TITLE: Advances in Vehicle Systems Concepts and Integration. [les Avancees en concepts systemes pour vehicules et en integration]

To order the complete compilation report, use: ADA381871

The component part is provided here to allow users access to individually authored sections of proceedings, annals, symposia, ect. However, the component should be considered within the context of the overall compilation report and not as a stand-alone technical report.

The following component part numbers comprise the compilation report:
ADP010300 thru ADP010339

UNCLASSIFIED

UNMANNED RESEARCH VEHICLE (URV): DEVELOPMENT, IMPLEMENTATION, & FLIGHT TEST OF A MIMO DIGITAL FLIGHT CONTROL SYSTEM DESIGNED USING QUANTITATIVE FEEDBACK THEORY

By

C. H. Houpis, Professor Emeritus, Air Force Institute of Technology
Senior Research Associate Emeritus, Air Force Research Laboratory

S. J. Rasmussen, Research Associate, Air Force Institute of Technology

Air Force Institute of Technology/ENG, Bldg 642, 2950 P Street
Wright-Patterson Air Force Base, OH 45433-7765, U.S.A.

Abstract

The Quantitative Feedback Theory (QFT) design technique, which has the ability to bridge the gap between theory and the real-world control design problem, is utilized in the design of a MIMO digital flight control system for an unmanned research vehicle (URV) that is presented in this paper. The design illustrates how the "real-world" knowledge of the plant to be controlled and the desired performance specifications can be utilized in trying to achieve a successful robust design for a nonlinear control problem. This paper presents some of the issues involved in developing, implementing, and flight testing a flight control system (FCS) designed using QFT. Achieving a successful FCS involves a number of steps: specification of the control problem, aircraft model data, theoretical flight control system design, implementation, ground testing, and flight test. The last three steps embody the "practical engineering" aspects that are vital to achieving a successful FCS. The main emphasis of this paper is on these steps. First, there is a brief explanation of the MIMO design QFT process. This is followed by a description of the steps involved in the implementation and testing of a QFT designed FCS. Thus, this presentation provides an overview of "using robust control system design to increase quality" in attempting to demonstrate the "Bridging the Gap" between control theory and the realities of a successful control system design. In facing the technological problems of the future, it is necessary that engineers of the future must be able to *bridge the gap*, i.e., this "Bridging the Gap" must be addressed to better prepare the engineers for the 21st century.

I INTRODUCTION

In facing the technological problems of the 21st century, it is necessary that engineers of the future must be able to *bridge the gap* between the scientific and engineering methods. Developing a set of Engineering Rules (E.R.) is a first step towards achieving this goal (see Chap. 9 of ef. 1). This paper provides the next step in enhancing this goal: overcoming problems encountered during design, implementation, and achieving a successful real world QFT designed FCS control system. The QFT technique is a design method that has the inherent capability to assist in bridging the gap between the scientific and engineering methods. Thus, a discussion of the development, implementation, and successful flight test of a flight control system, designed using QFT techniques, is presented in this paper. The robust flight control system was designed for and flown on the Lambda Unmanned Research Vehicle (URV. Lambda is a remotely piloted aircraft that is operated by the Air Force Research Laboratory at Wright-Patterson AFB, OH for research in flight control technology.

Control design problems generally involve real world nonlinear plants. In utilizing control system design techniques, which require linear plant models, it is necessary that assumptions be made that allow simplification of these nonlinear plants, i.e., "assume linear behavior" that result in obtaining linear plant models. Thus, it is important for the designer to follow a design and implementation process that allows the testing of the assumptions as early in the process as possible so the control system can be redesigned, for example, to take into account unmodeled effects. As detailed in this paper, the control design process should include simulation of the control system on increasingly realistic models which helps transition to implementation on real world applications. Most of the real world implementation problems are the result of assumptions made during the design process.

II OBJECTIVE

The objective of this project was twofold. First, develop a robust flight control system using QFT, and take the de-

Paper presented at the RTO SCI Symposium on "Warfare Automation: Procedures and Techniques for Unmanned Vehicles", held in Ankara, Turkey, 26-28 April 1999 and published in RTO MP-44.

sign through a flight test. Second, implement an inner loop FCS on the Lambda URV that would be part of an autonomous flight control system. During the project the first objective was accomplished and then, because of hardware improvements, a second design was developed and flight tested. This second design was accomplished to better meet the requirements of the second objective. The FCS design process used is shown in Fig. 1. As indicated by the arrows in the one complete FCS design cycle covers the process through the flight test and then back to the re-design stage. During this project there were four cycles around this loop. Two of the cycles produced unsuccessful flight tests and two produced successful flight tests.

III. QFT DESIGN PROCESS[1-4]

The QFT technique requires that $\iota = 1,2,3,\ldots,J$ LTI models be determined that represent the dynamical model over its operating scenario in order to achieve a robust design. These LTI plants determine the template contours which represent the region of plant parameter uncertainty and are used in the QFT design technique. The robust digital flight control system design was performed as a psuedo-continuous-time (PCT) control system. Upon completion of the design the compensators and prefilters are transformed into the z-domain controllers and prefilters by use of the Tustin transformation.

IV CONTROL SYSTEM DESIGN PROCESS (FIG. 1)

In order to design a control system for a real world control problem, the designer must follow a design process such as that shown in Fig. 1. This figure represents a design process that moves the designer from the problem definition stage to the successful control system implementation in steps of increasing reality. If the control system does not meet performance specifications at any stage of the process, the control system is redesigned and retested. In general, as the simulations become more realistic, they also become more expensive both in cost and time. Therefore, it is very important to be able to find potential problems early in the design process for the control system. The ovals inside the circle in Fig. 1 indicate the features of the QFT technique that assist in the design of control systems and can best meet performance specifications and be implemented on the real world system. The following sections describe the individual stages of the control design and implementation process. Indicated in the following sub-section titles is a number that refers to the block number in Fig. 1 to which the sub-section applies.

III-1 FUNCTIONAL REQUIREMENTS (#1)

The designer, at the onset, must have a clear understanding of the problem that needs to be solved. That is, the designer must understand what the controlled system is required to do and what are its operational requirements. The designer must also understand the environment in which the system is required to operate, i.e., the environmental requirements. Together these two requirements make up what is referred to as the *functional requirements*. If the designer does not start with a clear understanding of the functional requirements, costly time can be wasted in the design-test-redesign cycle. If during the design process, it becomes clear that the functional requirements cannot be met, the designer might be called upon to use engineering judgement and the knowledge of the goals of the controlled system to modify these requirements. Note, this is not a step that a control designer normally takes on his own.

III-2 PERFORMANCE SPECIFICATIONS (#2)

Performance specifications[1,2] are essentially mathematical models developed from the functional requirements and are utilized during the design process in order to achieve the desired system performance robustness. Since performance specifications are normally only interpretations of the functional requirements, the designer must be aware of how the specifications and requirements relate and what tradeoffs need to be made. During the design process, the designer might need to apply engineering judgement in order to make the necessary modifications to the specifications that, while still meeting the requirements, enables achieving a robust control system design.

III-2.3 DYNAMICS MODEL (#3)

A *dynamic model* is a mathematical model of the system to be controlled and is developed from a knowledge of the system and its operating requirements. This model can be as simple as a linear-time-invariant (LTI) transfer function or a complicated set of nonlinear differential and algebraic equations with time varying parameters. In many cases, a simplified model of the dynamical system can be used to represent the system in the control design process. In fact, the designer should try to use as simple a model as possible that represents the important system dynamics in the design process. For example, from an analysis of the LTI transfer functions a designer may be able to determine their nondominating poles and zeros, i.e., those which have a negligible effect on the system's performance (those that lie outside the system's bandwidth). Thus, by deleting the nondominating poles and zeros from these LTI transfer functions reduced order models are obtained. Not only does a reduced order model simplify the design process, but also reduces the risk of introducing numerical inaccuracies in the design process. But remember, an oversimplified model can lead to trouble as in the case of bending modes as discussed in Sec. IX

III-2.4 CONTROL AUTHORITY ALLOCATION (#4)

An important part of the design process is the *control authority allocation* assigned to each of the control effec-

tors. Depending on the dynamical system, there may be redundant control effectors, i.e. the number of control effectors available to the controller may be greater than the number of controlled variables. Also, the control effectors available may induce cross-coupling in the dynamical system and do not clearly control any one variable. In these cases, judgement must be exercised by the designer, based upon knowledge of the real-world operating characteristics of the plant, in determining the percentage of the control authority that is allocated to the various controlled variables. That is, a method for determining the percentage of control power available from each control effector to each controlled variable must be determined. The optimization of the control effectors' control authority allocation can be used to help decouple the system and assist in achieving the desired robust system performance. This control authority allocation is accomplished by the proper selection of the w_{ij} elements of the weighting matrix $\boldsymbol{W}$.

III-2.5 QFT CONTROL SYSTEM DESIGN (#5)

The QFT design process is used to develop mathematical algorithms that can be implemented in order to achieve the desired control system performance. Implementation issues and insights provided by the QFT process to the designer are discussed in the following sections. A QFT design can be accomplished by use of the MIMO QFT CAD package[5] which greatly simplify the design process.

III-2.6 LINEAR SIMULATION (#6)

Once the control algorithms have been designed, they are implemented along with linear representations of the dynamical system. These systems are simulated and the results are compared to the specifications. Since QFT design involves linearizing non-linear equations, the control system must be simulated for each of the J LTI transfer functions to check the result against the specifications. If some or all of the specifications have not been met, the designer can either redesign the control system or reexamine the requirements. In some cases, the initial specified requirements may not be realistic. For designs that involve control effector damage, the designer must ensure that the assumed percentage of effector damage is realistic with respect to its associated remaining control authority available to satisfy the control system performance requirements; for example, to still be able to fly the aircraft. Also, the designer must ensure that the system performance is close enough to the specifications to meet the overall functional requirements.

III-2.7 NONLINEAR SIMULATION (#7)

Once the control system has passed the linear simulation testing phase, the simulation complexity is increase by adding nonlinear components and any other components that are removed to simplify the simulation. As with the linear simulations it may be necessary to accomplish a redesign or a revaluation of the specifications (performance specifications, control authority allocation, and/or the percentage of control effector failure).

III-2.8 ENGINEERING VISUALIZATION (#8)

After each of the simulations it is valuable to animate, by a computer simulation, the dynamics data to better understand exactly what occurs during the simulation. Note that the three dimension engineering visualizations integrate all of the dynamics of the simulation. For example, in the case of an aircraft (A/C) this means that the designer can view the angle of attack, pitch rate, pitch attitude, forward velocity, vertical velocity, and altitude simultaneously. Instead of trying to decipher the position and attitude of the A/C from six two dimensional plots, the designer can obtain a clearer understanding from watching the computer animation of the maneuver. For more specific details of the maneuver the designer can then return to the data plots.

III-2.9 ENGINEERING INTERACTIVE SIMULATION (#9)

When there is an operator involved in the controlled system, for example, a pilot flying an A/C, it is often useful for the designer to use an interactive simulation in order to obtain a better understanding of the operation of the system. It should be noted in reality that the pilot is a part of the overall flight control system, i.e., he forms the "outer loop" of the control system. Thus, this type of control system is referred as a *manual flight control systems*. An interactive simulation provides the designer with the ability to implement the control system in the same fashion that it will be implemented on the dynamical system. The interactive simulation also gives the designer the ability to test the system continuously throughout the operating environment. In the case of a control system designed for an A/C, the interactive simulation involving a pilot gives the designer the ability to perform a simulated flight test before the design leaves her/his desk. Such simulations, for a specified A/C, are often performed by a pilot, for example, at the Wright-Patterson AFB Lamars simulator.

III-2.10 HARDWARE-IN-THE-LOOP SIMULATION / IMPLEMENTATION (#10)

At this stage of design and implementation the control system algorithms are implemented on the same type of hardware systems as those that control the dynamical system. Other hardware components such as actuators and sensors are also connected to the system. This allows simulation of real-time operation of control algorithm, noise corrupted measurements for feedback, and computation cycle time/sampling rate quantization errors. A hardware-in-the-loop-simulation is also useful to ensure that commands issued from the control system move the effectors in the cor-

rect directions and the outputs of the feedback sensors have the correct polarity.

III-2.11 OPERATOR-IN-THE-LOOP SIMULATION (#11)

In order to insure the controlled system meets the requirements of the human operator a simulation is set up to allow the operator to interact with a simulation of the system. Many of these simulations surround the operator with visual cues and some, inject motion into the simulation. These types of simulations are used to improve the handling qualities of the controlled system by giving the operator a chance to try out the controlled system and then using his or her responses to help shape a redesign.

III-2.12 SYSTEM TEST (#12)

The final testing of the control system involves implementation on the dynamical system and operational testing. Once the controlled system has been shown to meet the performance specifications for the operating environment, a successful control design has been achieved.

III-2.13 REDESIGN (#13)

At every stage of the control system design and implementation process the designer makes a decision to move to the next stage or to redesign (modify) the control system. Once the control system is modified the simulation testing is repeated.

IV DESIGN PROCESS EXAMPLE

The Lambda Unmanned Research Vehicle (URV) shown in Fig. .2 is a remotely piloted A/C with a wingspan of *14 ft* and is operated by the US Air Force for research in flight control technology. The objectives of the project described in this section are as follows:

1. To design robust flight control systems using the QFT design technique
2. To flight test these designs
3. To implement an inner loop FCS on the Lambda URV that would be part of an autonomous flight control system
4. To illustrate some of the real-world problems that are encountered in performing the control system design process shown in Fig. 1.

In accomplishing this design project required four cycles around the control design process loop. These four design cycles are:

Cycle 1 – This cycle involved the satisfaction of only the first two of the project objectives.

Cycle 2 – Cycle 1 was repeated but involved the design of an improved integrator wind-up limiter.

Cycle 3 – A redesign of the FCS was accomplished to satisfy requirements 1 through 3.

Cycle 4 – A refinement of the plant model was made in order to take into account a bending mode that was neglected in the previous designs.

Cycles 1 and 3 were unsuccessful and cycles 2 and 4 produced successful flight tests.

IV-1 FIRST DESIGN CYCLE

Requirements

There were two major design requirements for this project. The first was a desire to develop a robust flight control system using QFT, and take the design through flight test. The second was a need for an inner loop FCS on Lambda that would interface with an autonomous waypoint directed autopilot.

Specifications

The time response specifications were selected base on the open-loop response of Lambda. The pitch rate was an underdamped response that settled fairly quickly. Overshoot and settling time were chosen to be *25%* and *1 sec.*, respectively, for pitch rate response. Roll rate was an overdamped response that settled quickly, and the settling time was chosen to be one second. Yaw rate was also underdamped, but it did not reach steady state as fast as the other two. Yaw rate overshoot and settling time were chosen to be *15%* and *2 secs.*, respectively. These specifications were transformed into LTI transfer functions for use in the QFT design.

Aircraft (A/C) Model

The A/C model developmental process began with the use of Digital Datcom, a computer program which predicts stability and control derivatives for aerospace vehicles based on the physical characteristics of the vehicle. Datcom information forms the baseline model of the A/C. This baseline model was refined by using system identification software to estimate the aerodynamic derivatives from actual flight test data[6]. Maximum likelihood identification was used to identify the natural frequency and damping ratios of the short period and roll modes. This information combined with the Datcom information provided a working model for the flight control system design.

FCS Design

There were two QFT designs accomplished at the Air Force Institute of Technology[7,8] (AFIT) The first was based on the DATCOM model of Lambda alone. The second design was based on the DATCOM model with the refinements made

with system identification. This second design used linearized transfer functions to represent Lambda in various flight conditions, covering the entire proposed flight envelope, to accomplish the design and for linear simulations.

Linear Simulations and Nonlinear Simulations

All FCS designs were simulated using Matrix_x and LTI state space models representing the full flight envelope of Lambda. After successful linear simulations, nonlinearities such as control surface travel limits were introduced into the linear simulation. A nonlinear simulation was developed at the Air Force Research Laboratory (formerly the Wright Laboratory) that incorporated a six degree of freedom simulation, automatic trim calculation, air vehicle kinematics, and control surface saturation. While this design produced the desired responses in the linear simulation, when implemented in the nonlinear simulation the original control system exhibited undesirable behavior due to the initial assumptions about allowable gain being incorrect. Thus, the allowable gain was modified to achieve a redesigned controller.

Hardware-in-the-Loop Simulation

Software from the nonlinear simulation were used to develop a hardware-in-the-loop simulation[9]. This simulation allowed the implemented FCS, which is programmed on a EPROM chip, to be tested in the A/C. When the FCS was implemented in this simulation, it was discovered that the angular rate sensors had high levels of noise, with peak values on the order of *0.5 deg/sec*. The FCS amplified this noise and this effectively masked any control command signal. The noise was recorded and was incorporated into the nonlinear simulation. The MIMO QFT CAD[1,2,5] for designing control systems allows for a rapid redesign. The noise problem was minimized by lowering the loop transmission gain and then testing the resulting FCS in the nonlinear simulation. This remedy was an "engineering decision" in order to obtain a satisfactory design. In the Third Design Cycle a more satisfactory resolution of the noise problem was achieved. Once simulations of the redesign were satisfactory, the FCS was flight tested (Flight Test #1).

Flight Test #1

Two major difficulties caused the first flight test to fail; the first was reversed polarity on an angle sensor and the second was an integrator wind-up limiter scheme that did not work. Since the inner loop FCS was to be implemented as a part of an autonomous system, turn coordination logic was implemented around the inner loop FCS that relied on the roll angle. Post flight analysis of the flight test video and data showed that the polarity of the roll angle sensor was backward, thus, when the A/C was commanded to bank, the rudder was commanded to deflect in the wrong direction. The FCS was thus turned off and the testing involving the lateral control channel was terminated. Later, during the same flight test, when the FCS pitch channel was turned on, the aircraft developed a high pitch rate. This test was also terminated and post analysis revealed that the scheme used to limit integrator wind-up had caused a numerical instability.

IV-2 SECOND DESIGN CYCLE

Requirements and Specifications and Aircraft Model

The requirements for the second design cycle did not change from the original requirements. An additional requirement was incorporated for the second design cycle that involved the design of an improved integrator wind-up limiter. The specifications and the A/C model for the second design cycle did not change from the original requirements.

FCS Design

Since the problems encountered in the first test had nothing to do with the QFT designed FCS, the same QFT FCS designed for the first flight test, was used in the second flight test. During the second flight test, there was no attempt to use a turn coordination algorithm. The insertion of an integrator wind-up limiter involved a different form of the controller implementation for the second design cycle. In this cycle instead of each of the controllers being implemented by a single software algorithm relating their respective outputs to their respective inputs, they were implemented in the manner described by E.R.12 of Chap. 9 of Ref. 1. That is, the continuous time domain transfer functions were factored into poles and zeros in order to create first order cascaded blocks (transfer functions) that were individually transformed into the discrete time domain. The individual transfer functions were then implemented, by their own respective software algorithm.. This implementation allowed limitations to be placed only on those pieces of the FCS that contained pure integrators and provided the required controller accuracy.

Linear, Nonlinear, Hardware-in-the-Loop Simulation

All simulations consisted of checking out the new implementation of the FCS. There were no problems encountered during any of these simulations.

Flight Test #2

On 20 Nov 92, the temperature was in the *60°F+* with winds at *5* to *7 mph*. Lambda was flown in manual mode for takeoff, setup, and landing. Due to problems with the first flight test the FCS was engaged only during the test maneuvers. The maneuvers performed consisted of unit step commands in all three axes. This set of maneuvers was first performed with the QFT FCS and then with the open loop A/C. As shown in Fig. 3, the QFT FCS performed as it was designed. The figure shows the responses of Lambda to a step pitch down command. The dotted lines in the plot represent the

specified T_{R_U} *and* T_{R_L}. It is important to note that during this maneuver the A/C covered a large portion of its dynamics envelope by varying in forward airspeed from *75 kts* to *110 kts*.

IV-3 THIRD DESIGN CYCLE

Requirements

The requirements for the third design cycle had not changed from the original requirements. This cycle involved the design of an inner loop FCS that had intrinsic turn coordination. Also, the sensor noise problem was reduced by an order of magnitude by the addition of a hardware noise filter on the output of the sensors. It was determined that the noise originated from a motor on the sensor; the noise was a high frequency noise that was being sampled at a lower frequency. Thus, this aliased noise had a relatively high bandwidth. The remedy was to place a filter at the sensor output before the sampler. This allowed a redesign of the FCS to improve the system performance.

Specifications

For this iteration of the design a *sideslip angle command* was incorporated as part of the inner loop controller. Since Lambda has a sideslip sensor, a sideslip command was used to cause the A/C to intrinsically fly coordinated turns. That is, the goal of turn coordination is to reduce the sideslip angle to zero during a turn by using the proper amount of rudder deflection during the turn. Changing to sideslip command allowed the use of the yaw rate sensor to implement a *yaw damper* to reduced the dutch roll mode oscillations. This yaw damper was implemented by adding a washout filter, designed through the use of a root locus plot. The yaw damper was designed and then incorporated in the A/C model for a FCS design. During the second flight test the pilot felt that the aircraft's roll rate response was too slow. Therefore, the roll rate response specification was change to match that of the pitch rate. After this change the roll specifications for overshoot and settling time were *2.5%* and *1 sec*, respectively.

Aircraft Model

The sensor improvement, mentioned above, was included in the nonlinear aircraft model by recording actual noise and inserting it as a block in the model. During the system identification work for the second A/C model, some of the parameters had been scaled incorrectly. This caused some modeling errors. After the second flight test these errors were corrected through the use of system identification applied to flight test data that resulted in a refinement of the A/C model.

FCS Design

Matrix$_x$ was used to develop linearized plant models about flight conditions in the flight envelope. An attempt was made to choose flight conditions in such a way as to fully describe the flight envelope with the templates. To do this a template expansion process was developed and is explained in Sec. V.

Linear, Nonlinear, and Hardware-in-the-Loop Simulations

The refined Lambda model was implemented in all three simulations. The FCS was implemented in the cascaded method outlined previously. All simulations produced the desired responses to given stimulus.

Flight Test #3

During the third flight test, when the FCS was engaged, the A/C exhibited an uncontrolled pitching, or porpoising, behavior. While the post flight test analysis was inconclusive, a longitudinal bending mode at *13.2 rad/sec* seemed to be the likely cause.

IV-4 FOURTH DESIGN CYCLE

Requirements and Specifications

The requirements for the fourth flight test had not changed from the original requirements, but involved a refinement in the aircraft model to incorporate effects of the bending mode discovered in Flight Test #3. The specifications for the fourth design cycle were the same as those for the third cycle.

Aircraft Model

A model of the porpoising behavior encountered in the third flight test was identified by assuming that the behavior was caused by an unmodeled effect. Various models were incorporated into the nonlinear model and simulated. This simulation used the identical flight test inputs as simulation inputs and compared the simulated outputs to the flight test data. Using this procedure, see Sec. VIII, a violation of the gain margin was ruled out by increasing the inner loop gain in the model and observing the response. Instability caused by actuator rate limiting was ruled out by inserting severe rate limited actuator models in the nonlinear simulation. When a bending mode, modeled as a lightly damped pair of poles, was inserted in the model, the simulated responses were very similar to the flight test results.

FCS Design

Matrix$_x$ was used to develop linearized plant models about the given flight conditions and the FCS was redesigned based on the model containing the bending mode. Note, when the FCS from design cycle three, using the A/C model with the bending mode, there were violations of stability

criteria in the frequency domain and, as expected, the porpoising behavior occurred.

Linear, Nonlinear, and Hardware-in-the-Loop Simulation

A fourth design cycle was accomplished using the new model. This design was implemented and all three simulations were run and tested. This FCS design simulation responded within specifications and, as expected, the porpoising effect was eliminated.

Flight Test #4

The fourth flight test occurred in September 1993. The field conditions were a little gusty, but within acceptable limits for the flight test. During the flight the FCS was engaged and then left engaged for the entire series of tests. The FCS performed as designed. The intrinsic turn coordination scheme worked as designed. The pilot was pleased with the handling qualities and felt comfortable flying with the FCS engaged at all times. His one criticism was that the roll rate was too slow. Since the roll rate was limited by the maximum roll rate detectable by the roll rate gyro, the problem was unavoidable. When the data was examined, it was found that all of the *60 Hz* data had been lost, but much of the *10 Hz* data had been captured. Analysis of this data showed that the FCS did cause Lambda to respond within the specified envelope, during onset of the command, but, in some cases, Lambda's response exhibited more overshoot and longer settling time than specified. These problems could be attributable to the gusty conditions, since no gust disturbance was specified during the design process. More flight testing of this FCS will be required to answer this question.

V SELECTION OF DESIGN ENVELOPE

At the onset of a QFT design, the designer must select a set of operating conditions in order to obtain the LTI transfer functions that represent the dynamical system and which are used to obtain the templates that are required for the design. The problem is which operating conditions to choose. Only those operating conditions that yield points that lie on the contour of the templates, for all frequencies of interest, are necessary. Choosing too many LTI plants may yield points that lie inside the template contours and can lead to computational problems during the design. Note by applying engineering insights it is readily determined that the template contours and not the LTI plants which lie within the template's contour determine the performance bounds that need to be satisfied by the synthesized functions. Thus, the computational workload and associated problems may be minimize by reducing the number of plants to be utilized in the design process to only those plants that lie on the template contours.

Through engineering knowledge of the problem the designer is able to determine the particular parameters that effect the operating conditions and the physical limits of these parameters. In the case of Lambda the parameters that were varied to set the operating conditions were airspeed, altitude, weight, and center of gravity. Gross limits were set for these values from knowledge of the A/C and the possible flight envelope. Next, the template expansion process was used to find the set of operating conditions that fully described the flight envelope. The template expansion process, shown in Fig. 5, is a graphical process that tracks the effect of variations of the parameters which are involved in selecting the operating conditions and determine the resulting LTI plants. The process is as follows:

1. Determine the important parameters that describe the operating condition and their minimum, maximum, and nominal values.
2. Choose a template frequency for the expansion process. This frequency should be representative of the dynamic system in the bandwidth of interest. At the end of the process, other template frequencies should be checked to insure that a complete set of operating conditions have been chosen.
3. For the template frequency of step 2, plot the dB vs phase values of the nominal operating condition.
4. On this same graph, plot the results of varying each parameter through its maximum and minimum while holding the rest of the parameters at their nominal values. This forms an initial template.
5. Identify the variations caused by each parameter. This can be accomplished by connecting the points on the template due to each parameter variation.
6. Choose the two parameters that cause the largest variations and use these to expand the template. This is accomplished by holding the remaining parameters at their nominal values and plotting the four points of the templates resulting from the extremes of the two parameters.
7. Use the outside points, on this expanded template, as nominal points for further expansion with other parameters.
8. Choose other frequencies in the bandwidth of interest to ensure that the operating envelope is completely defined.

For Lambda, a nominal flight condition was chosen to be *50 kts,* velocity, *1,000 ft* altitude, a weight of *205 lbs*, and center of gravity at *29.9%* of the mean aerodynamic cord. From this nominal trim flight condition, each parameter was varied, in steps, through maximum and minimum values, while holding the other parameters at their nominal trim values. These variations produced an initial set of templates. On these templates, the variation corresponding to each parameter was identified. Each variation when translated, on the template, identified an expanded template area of the flight envelope that required more plants for better definition.

VI CONTROL SYSTEM IMPLEMENTATION ISSUES

An implementation problem that can cause stability and performance problems is integrator wind-up. This is the situation that occurs when the controlled system cannot respond quickly enough to the commands from the controller and the commanded values keep increasing due to integrator action. A situation like this occurs when a control effector has reached its limits. The longer the system is in this state the more the commanded value increases. The problem occurs when the controller tries to reverse the command, the commanded value must be "integrated" back down to the operational range before it becomes effective. In order to prevent integrator wind-up, anti-windup algorithms must be applied to integrators during implementation. During the QFT design process the controller is in the form of transfer functions that can be of any order. For implementation, these transfer functions can be separated into first and second order transfer functions (see E.R. 12 of Chap. 9[1]). With the transfer functions separated in this manner individual integrators can be limited.

VII HARDWARE/SOFTWARE CONSIDERATION

During the modeling and development of the control system, assumptions were made as to the polarity of feedback and command signals. During implementation these assumptions must be tested. This is one of the reasons to use a hardware-in-the-loop simulation. With this type of simulation the control algorithms can be implemented and the control effectors can be monitored during simulated operation. Feedback signals can be checked by moving sensors by hand, if possible.

The other phenomena that a hardware-in-the-loop simulation can identify is the effects of feedback noise on the controlled system. If the feedback noise is within the bandwidth of the control system, and the noise has not been included in the modeling or simulation, the controller may need to be redesigned to account for the noise. This might result in a trade off between performance and noise rejection. Sometimes it is possible to implement a hardware filter after the sensor, but before the sampler too reduce the noise in the bandwidth of interest.

VIII BENDING MODES

During the design of a control system, the effects of higher frequency modes on stability and performance must be considered. In A/C, one source of higher frequency modes is structural bending. A control system that excites a bending mode in a flying A/C can produce disastrous consequences. During the modeling process it is very important to include the effects of these higher frequency modes so they can be minimized during the design process. In the case of Lambda, the existence of a bending mode was discovered during a flight test.

VIII-1 Lambda Bending Example

Following the initial flights, the A/C operators decided that they would prefer a different feedback structure in the FCS that included turn compensation. Thus, to implement turn compensation, a sideslip angle command was incorporated as part of the inner loop controller. The goal of turn coordination is to reduce the amount of sideslip angle during a turn by using the proper amount of rudder deflection during the turn. Since Lambda has a sideslip sensor, sideslip feedback was used to cause the A/C to intrinsically fly coordinated turns. Changing to sideslip command also allowed the use of the yaw rate sensor to implement a yaw damper to reduced the dutch roll mode oscillations. This yaw damper was implemented by adding a washout filter, designed through the use of a root locus plot. The yaw damper was designed and then incorporated in the A/C model for a FCS design.

When this design was finally flight tested, a porpoising behavior was observed. To ensure flight safety, Lambda was flown to a safe altitude by the pilot before the QFT FCS was engaged. The pilot had Lambda flying in level flight when the longitudinal portion of the QFT FCS was engaged. At this point Lambda began oscillations in the pitch axis and the QFT FCS was disengaged immediately. In order to collect sensor data on this behavior, Lambda was flown back to level flight, the longitudinal portion of the QFT FCS was engaged and the sensor data was recorded for further analysis. Pitch attitude data from this flight is shown in Fig. 5 whose high resolution data was at a 60Hz sample rate.

VIII-2 Unmodeled Behavior

A model of the porpoising behavior was identified by assuming that the behavior was caused by an unmodeled effect. Various proposed models were incorporated into a nonlinear model of Lambda and simulated. This simulation used the actual flight test inputs as simulation inputs and compared the simulated outputs to the flight test data. Using this procedure, a violation of the gain margin was ruled out by increasing the inner loop gain in the model and observing the response. Instability caused by actuator rate limiting was ruled out by inserting severe rate limited actuator models in the nonlinear simulation. Upon reviewing the video record of the flight, it was suggested that the A/C appeared to have a second-order bending mode in the longitudinal axis. It was possible to excite and observe such a mode by tapping rhythmically on the tail of the A/C.

A bending mode modeled as a lightly damped pair of poles at *13.2 rad/sec*, just within the bandwidth of the FCS, was inserted in the nonlinear simulation as shown in Fig. 6. This model generated a pitch acceleration signal from elevator deflection which was passed through the second order filter:

$$\dot{q} = \frac{-20}{s^2 + 5.28s + 174.2}\delta_{elev}$$

The simulated response was very similar to the flight test results. $Matrix_X$ was used subsequently to develop new linearized plant models containing the bending mode about the given flight conditions. The Bode plots of these models are shown in Fig. 7

The new plant models were entered in to the MIMO QFT CAD software. The FCS was redesigned based on the new models using the FCS from the previous design cycle as a baseline. The previous controller was:

$$g_{11}(s) = \frac{1093(s+8.5)(s+11)(s+3.9 \pm j2)}{s(s+2)(s+80)(s+36 \pm j48)}$$

The MIMO QFT CAD software showed that, with the old controllers, there were violations of the stability criteria on the Nichols chart.

The standard method of design would be to add a notch filter to keep the mode from becoming excited. The bending mode is close enough in frequency to the performance bandwidth of Lambda that care needs to be taken to design a controller that will be able to take advantage of the available bandwidth to deliver performance, stability, and disturbance rejection without exciting the bending mode. A standard notch filter would not take advantage of any beneficial dynamics at frequencies nearv the bending mode. It would also increase the order of the compensator. As an alternative, the inner loop filter was revised to compensate for the new information. It was also possible to design a fourth-order controller to replace the earlier fifth-order design, lowering the complexity of the controller instead of increasing it. The new controller was determined to be .

$$g_{11}(s) = \frac{125(s+1)(s+2.5 \pm j9.4)}{s(s+10)(s+35 \pm j35.7)}$$

A characteristic of a bilinear transformation is that, in general, it transforms an unequal-order transfer function ($n_s \neq w_s$) in the s-domain into one for which the order of the numerator is equal to the order of its denominator ($n_z = w_z$) in the z-domain. This characteristic must be kept in mind when synthesizing $g_i(s)$and $f_{ii}(s)$. Therefore, a nondominating s-domain zero at -150 is inserted in g_{11}.

With the MIMOQCAD program it was possible to shape the loop so that at *5 rad/sec* the loop intersected a point on the Nichols chart where the stability boundary and the performance boundary met. This was an optimal point for the loop to pass through given Lambda's performance bandwidth. The new A/C model was implemented in the nonlinear simulations and tested with both controllers. As expected, the resonance occurred with the FCS that was designed in Design Cycle #3. The FCS resulting from Design Cycle #4 responded within specifications. The new FCS passed a hardware-in-the-loop simulation and was scheduled for a flight test. During the next flight test, the field conditions were gusty, but within acceptable limits for the experiment. The QFT FCS was engaged and there was no noticeable oscillation. The pilot was very pleased with the handling qualities and felt comfortable flying with the FCS engaged for the entire series of tests. The only problems encountered were some roll performance problems which could be attributed to the windy conditions. Pitch response during this flight is shown in Fig. 8. Unfortunately, the test data recording function failed during the flight so that the only data available is low resolution data ($\pm 0.5°$) recorded at *10Hz*.

IX SUMMARY

Control design and implementation in the real world is an iterative process. Initial steps are performed with linear models that have been formulated with simplifying assumptions. After successful testing of the designed control system, based upon these simplified models, it is tested on increasingly realistic (nonlinear) models. At any point in the design process, if the control system does not meet performance and stability specifications, the control system must be redesigned and retested on the simplified models. This redesign is followed, once again, by testing on the nonlinear model (see Fig. 1). At every point of the design process the designer must be aware of test assumptions so engineering judgement can be use to help guide the design to a successful implementation and operation. The bottom line is that the controlled system must meet the requirements set out at the beginning of the process.

REFERENCES

1. Houpis, C. H. and S. Rasmussen, "Quantitative Feedback Theory (QFT): Theory & Application," Marcel Dekker, Inc., New York, August 1999.
2. D'Azzo, J.J., and C. H. Houpis, "Linear Control System Analysis and Design," McGraw-Hill, NY, 4th Ed. 1995.
3. Horowitz, I. M., "Quantitative Feedback Theory." Proceeding of the IEE, Vol. 129, Pt D, No. 6, 1982.
4. Horowitz, I. M., "Quantitative Synthesis of Uncertain Multiple Input-Multiple Output Feedback Systems," Int. J. of Control, Vol. 30, No. 1, 1979.
5. Sating, R. R., "Development of an Analog MIMO Quantitative Feedback Theory (QFT) CAD Package," MS Thesis, AFIT/GE/ENG/92J-04, Graduate School of Engineering, Air Force Institute of Technology, Wright Patterson AFB, OH, 1992.
6. Swift, Gerald A. Model Identification and Control System Design for the Lambda Unmanned Research Vehicle. MS Thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH, Sept. 1991.
7. Lacey, D. J., Jr., "A Robust Digital Flight Control System for an Unmanned Research Vehicle Using Discrete Quantitative Feedback Theory," MS Thesis, AFIT/GE/ENG/91D, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, Dec. 1991.

8. Wheaton, David G., Automatic Flight Control System for an Unmanned Research Vehicle Using Discrete Quantitative Feedback Theory. MS Thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH, Dec. 1990.

9. Robertson, Scott D., A Real-time Hardware-in-the-Loop Simulation of an Unmanned Aerial Research Vehicle. Technical Report WL-TR-93-9005, Wright Laboratory-Wright-Patterson AFB, OH, Aug. 1992.

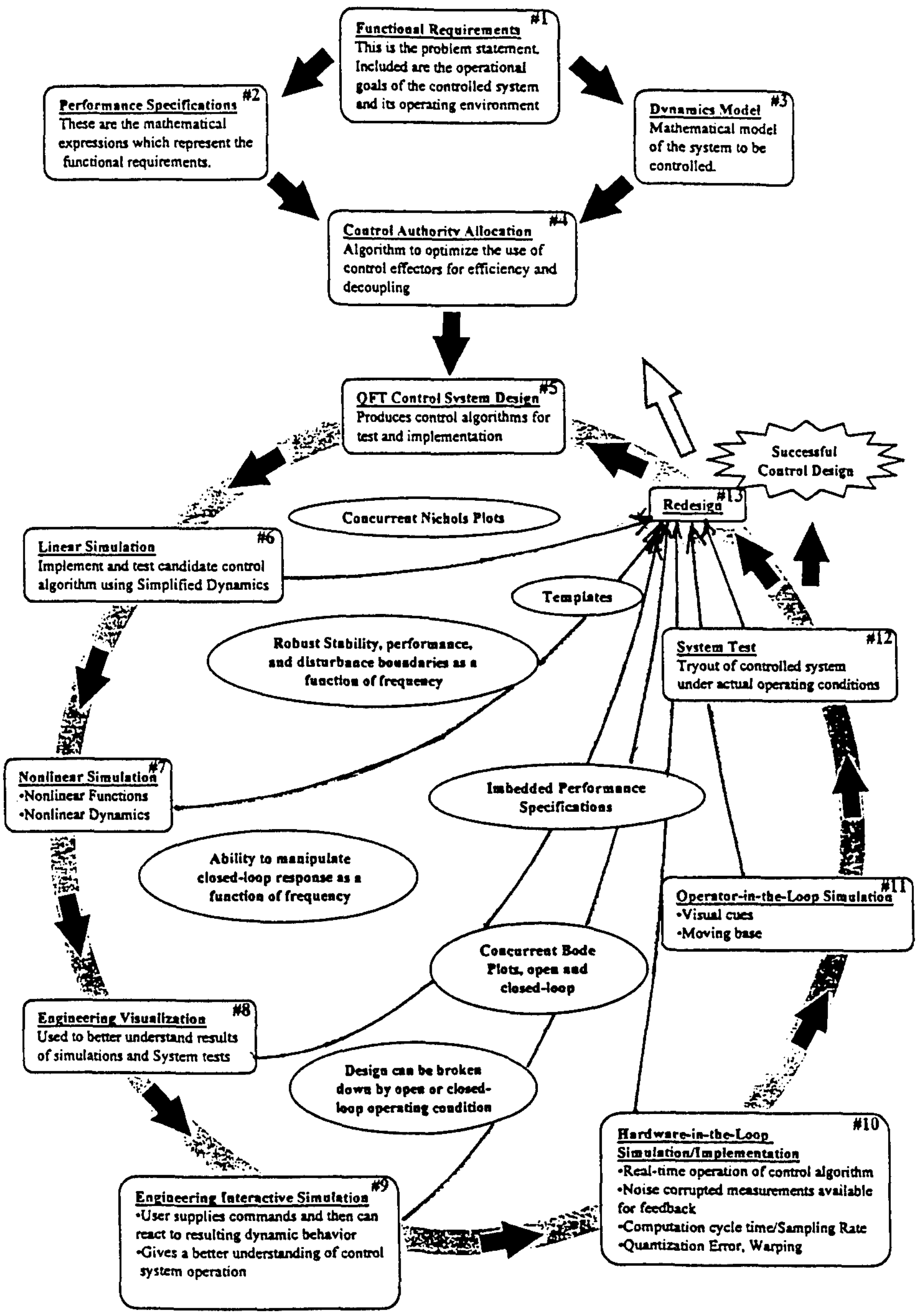

Fig. 1 The QFT control system design process: Bridging the Gap.

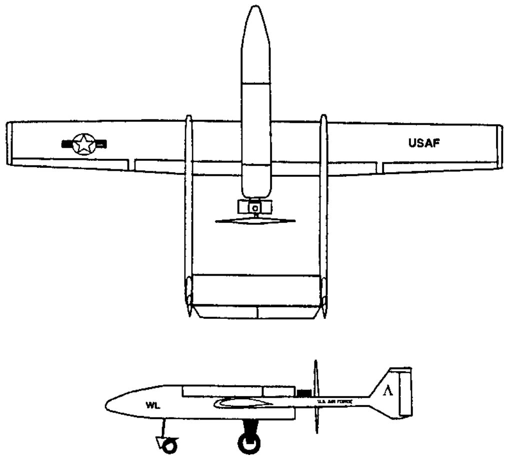

Fig. 2 Lambda Unmanned Research Vehicle (URV).

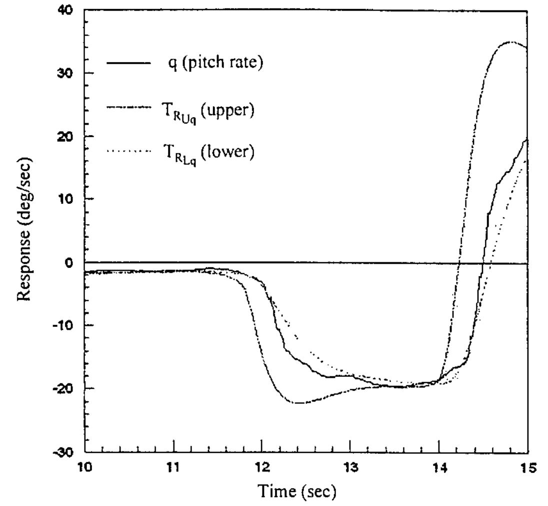

Fig. 3 Response to pitch-down command.

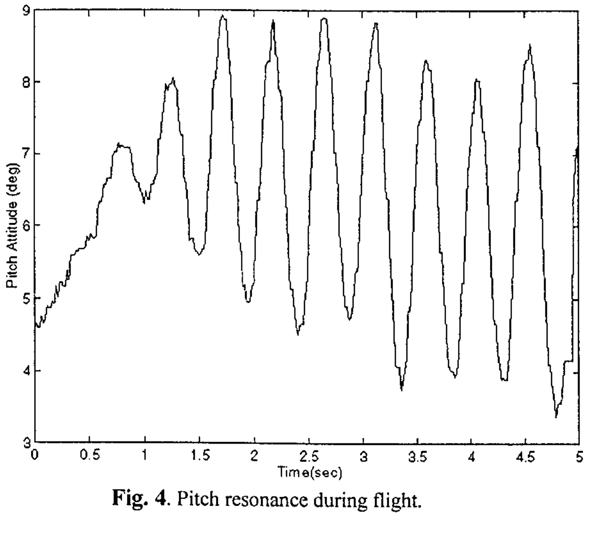

Fig. 4. Pitch resonance during flight.

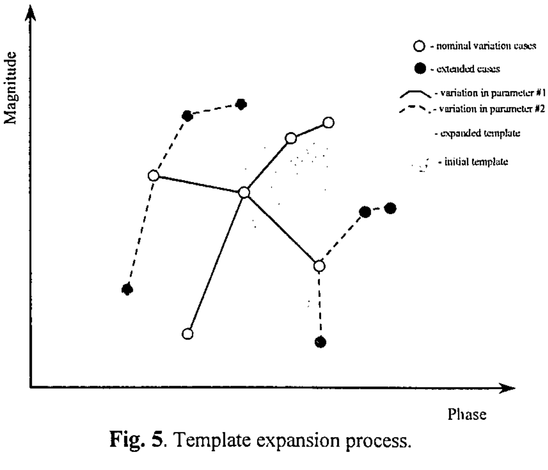

Fig. 5. Template expansion process.

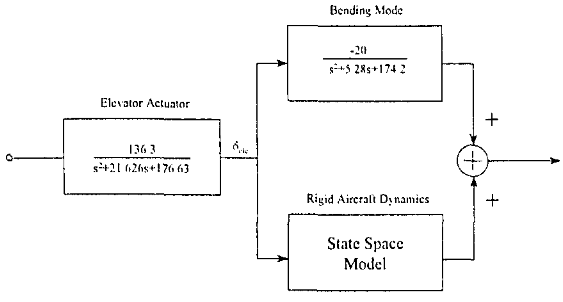

Fig. 6. Lambda bending model structure.

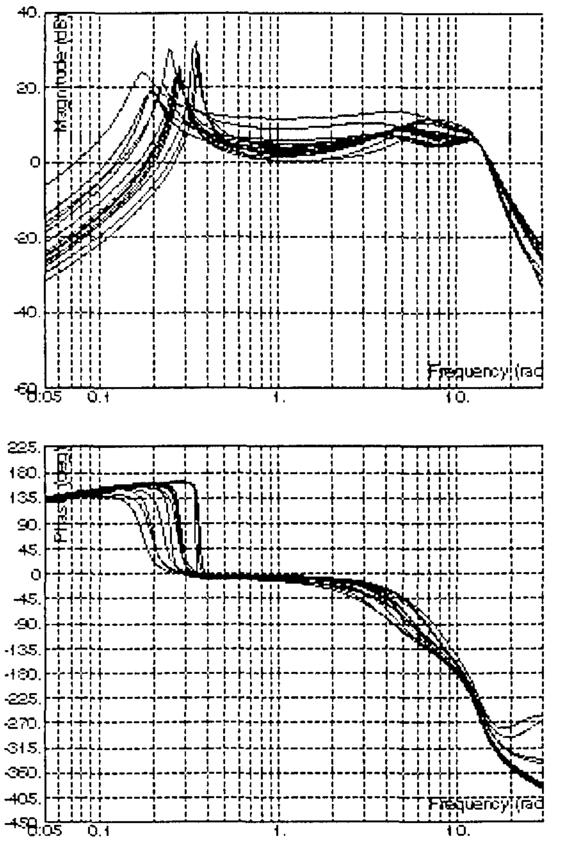

Fig. 7 Lambda bending models.

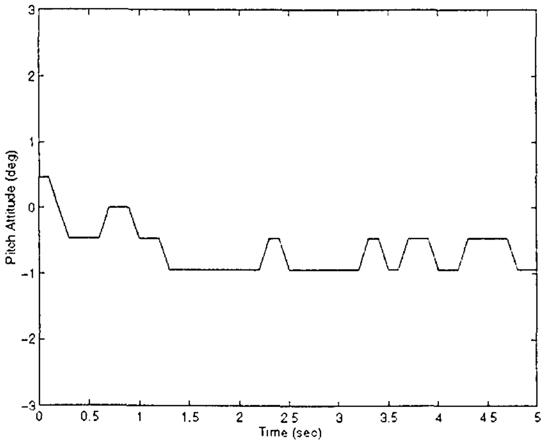

Fig. 8. Pitch attitude during flight.